Scalable Video Transmission Using Double Binary Turbo Code

Naeem Ramzan and Ebroul Izquierdo

Queen Mary, University of London

Mile End Road, London E1 4NS, UK

E-mail: {naeem.ramzan, ebroul.izquierdo}@elec.qmul.ac.uk

ABSTRACT

In this paper, we propose a novel efficient scheme for robust video transmission over wireless channels. The schema consists of motion compensated spatio-temporal wavelet decomposition scalable coder and double binary turbo code. The scalable bitstream is adapted with unequal error protection according to the relevance of video layers in the scalable bitstream. Cyclic redundancy check bits are also added to check error free delivery of scalable bitstream. Double binary turbo codes were used for protection since they perform better than classical turbo coders in terms of better convergence for iterative decoding and computational cost. Selected results of experimental evaluation are reported in the paper.

Index Terms— Scalable video coding, turbo code, joint source and channel coding

1. INTRODUCTION

In traditional video communications over heterogeneous channels, the video can be processed offline. Compression and storage is tailored to the targeted application according to the available band-width. However, keeping several different versions of encoded video in the servers is not a storage efficient solution. Furthermore, video delivery over heterogeneous and wireless channels meets challenges that are quite different from conventional transmission solutions. Here, dimensions like scalability, robustness and error resilience need to be re-defined to allow for variability according to individual user preferences and terminals.

Scalable coding promises to solve this problem by "encoding once and decoding many". The goal is to develop novel and efficient coding algorithms enabling content organization in a hierarchical manner to allow coding and interactivity at several granularity levels. Basically, scalable coded bit streams can adapt to the application requirements. Thus, problems inherent to channel capacity variations and better quality of services can be overcome. In wireless applications an additional critical problem relates to the very high bit error rate (BER). The challenge is to design efficient source and channel coding technology, so that the video content can be transmitted and used at different resolutions or quality levels.

In this work, a joint source-channel coding (JSCC) technique [1] is described. It combines the scalable video coding (SVC) framework reported in [2] with a forward error correction method (FEC) to achieve jointly source-channel scalable coding. Turbo codes (TCs) are one of the most prominent FEC techniques having received great attention since their introduction in 1993 [3]. This is mainly due to their excellent performance at low bit error rates, reasonable complexity, and versatility for encoding blocks with various sizes and rates. The JSCC technique proposed in this paper is based on SVC and double binary turbo coding [4].

The remaining paper is structured as follows: in section 2, double binary TC, is described. The proposed technique for JSCC is described in section 3. Section 4 presents some selected results. The paper closes with related discussions and conclusions in section 5.

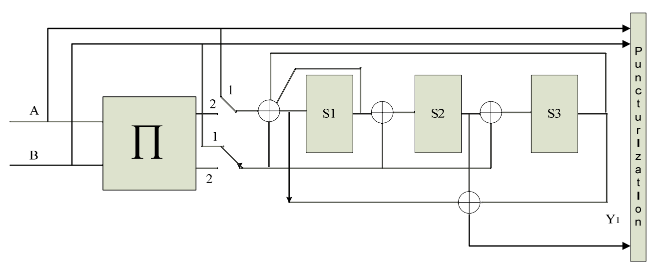

Figure 1: Double binary turbo encoder.

2. DOUBLE BINARY TURBO CODES

Double binary TCs were introduced in the domain of TCs by Berrou et al. [4]. These codes consist of two binary RSC encoders of rate $2/3$ and an interleaver of length *k*. Each binary RSC encoder encodes pair of data bits and produces one redundancy bit, so desired rate $1/2$ is the natural rate of the double binary TC. In this article, we consider the 8-state double binary TC with generators in octal notation are (15,13) for Y_1. This schema has been adopted by the European Telecommunications Standards Institute (ETSI) for Digital Video Broadcasting with Return

Channel via Satellite (DVB-RCS) and Digital Video Broadcasting with Return Channel via Terrestrial (DVB-RCT), as shown in the Figure 1. The tail-biting technique [4] is used to convert the convolutional code to block code that allows any state of the encoder as the initial state. So there is no need to tail bits to derive the encoders to the all-zero state. The interleaver design is a critical issue since the performance of the TC depends on how well the information bits are scattered by the interleaver. The used turbo-decoder is composed of two Maximum A Posteriori (MAP) decoders, one for each stream produced by the singular RSC block as shown in Figure 2. The first MAP decoder receives the two distorted systematic bits $(A_k^{'}, B_k^{'})$ after channel along with the parity $y_{k1}^{'}$ for first binary RSC encoder. Then it produces the *extrinsic information* Z_{k1} that is interleaved (Π) and feed to the second MAP decoder as the *a priori* information. The second MAP decoder produces the *extrinsic information* Z_{k2} based on interleaved distorted systematic bits $(A_k^{'}, B_k^{'})$, distorted parity by second binary RSC encoder $y_{k2}^{'}$ and *a priori* information from first MAP decoder. Then Z_{k2} is used as the *a priori* information of the first MAP decoder. After a certain number of iterations, usually 3 to 10, the *a posteriori probability* (APP) is taken, deinterleaved $(\Pi^{'})$ and finally a hard decision is performed to get the transmitted information. The robust version of MAP algorithm is Max-log-MAP [5].

At low error rates or high signal to noise ratio, the performance of the classical TC fluctuates due to the "error floor". The higher minimum distance can reduce the error floor effect at low error rates. Double binary TCs normally has better performance than classical TC due to larger minimum distance. The minimum distance of TC depends on the interleaver design or how well it shuffles the information bits. The used interleaver is based on the permutation equation $i = \Pi(j)$ where $j = 0...N-1$ and N be the number of data couples. The permutation is done on two levels; the first one inside the couple as $(A_j, B_j) = (B_j, A_j)$ if $j \bmod .2 = 0$

and second one between couples i.e.

$i = (P_0 \times j + P) \bmod .N$, with

-If $j \bmod .N = 0$, then $P = 0$.

-If $j \bmod .N = 1$, then $P = N/2 + P_1$.

-If $j \bmod .N = 2$, then $P = P_2$.

-If $j \bmod .N = 3$, then $P = N/2 + P_3$.

Where the values of P_0, P_1, P_2, and P_3 depend on N.

To get better performance for the double binary TC for video communication, the particular block length is selected that behaves better in the low error rates.

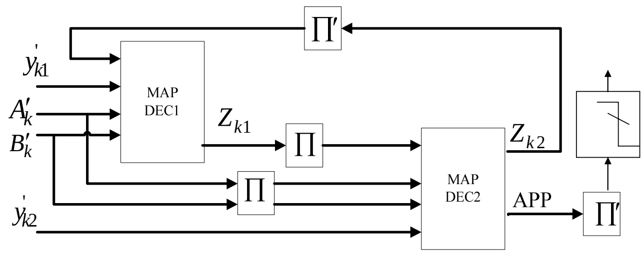

Figure 2: Iterative Turbo Decoding based on MAP algorithm for double binary TC.

3. PROPOSED TECHNIQUE FOR JSCC

Our scalable video codec [2] is based on a motion compensated temporal filtering (MCTF) and of spatial transform based on wavelets, producing a set of spatio-temporal subbands and $3D$ embedded entropy coding. The $t + 2D$ subband decomposition enables temporal and spatial resolution scalabilities, respectively. Spatio-temporal decomposition followed by a $2D$ or $3D$ embedded entropy coding leads to fine granular quality scalability on all supported spatial and temporal resolutions. The input video is encoded, producing the bitstream of the maximum required quality which can be extracted up to quasi-lossless level. The main features of the codec are: hierarchical variable size block matching motion estimation, scalable motion vectors, flexible selection of wavelet filters for both spatial and temporal wavelet transform on each level of decomposition, *2D* wavelet transform in lifting implementation and embedded zero-tree block entropy coder.

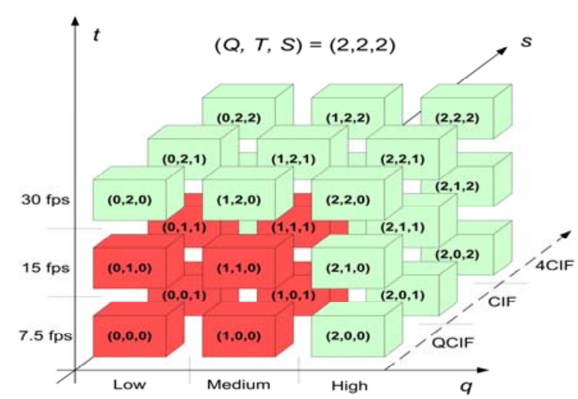

Figure 3: 3D representation of a scalable video bitstream.

The used scalable video bitstream consists of packets of data called atoms. An atom represents the smallest entity that can be added or removed from the bitstream. For easier interpretation, the bitstream can be represented in a $3D\ (q - t - s)$ space where *q*, *t* and *s* represent the quality, temporal resolution and spatial resolution. This space is outlined in Figure 3. In this figure, $Q = T = S = 2$. *Q*, *T*, *S* are the number of refinement layers in quality, temporal and spatial domain, respectively. There exists a base layer in each domain that is referred as 0^{th} layer except from refinement layers and cannot be removed from the bitstream. Therefore, in the example shown in Figure 3 we have 3 quality, 3 temporal and 3 spatial layers. Each atom has its own coordinates in $q-t-s$ space, which are denoted by (q,t,s). High level description of an actual scalable video bitstream is illustrated in [2], [6].

As there are several progression orders for multi-domain scalable bitstream, specifically for our case of quality, temporal and spatial scalability there are 3! = 6 orders. These are: QTS, QST, TQS, TSQ, SQT, STQ, where the convention is that the first letter refers to the domain which progresses most slowly, while the last to the one which progresses most quickly. Since in our codec the QTS progression is set as the default one, mapping $x \to (q,t,s)$ is defined by:

$$s = x \,\%\, (S+1)$$
$$t = \left\lfloor \frac{x}{S+1} \right\rfloor \%\, (T+1) \qquad (1)$$
$$q = \left\lfloor \frac{x}{(S+1)\cdot(T+1)} \right\rfloor \%\, (Q+1),$$

where % denotes modulo division and $\lfloor . \rfloor$ represents integer value of a positive real number [6].

If we select particular subsets of atoms regarding desired quality layers, frame rate and resolution, the selected atoms are interlaced in the embedded bitstream. Then the bitstream is divided into parts regarding the domain that progress most slowly. For example using the default setting of the SVC, the quality layers progress most slowly. Then we packetize each quality layer using packets of equal size. During the packetizing of the scalable bitstream, cyclic redundancy check (CRC) bits are added to check the error free delivery of packets at decoder side. Each quality layer is protected by FEC.

We use the double binary TC as FEC to adopt scalable video bitstream. Double binary TC's bit rate is twice at the decoder output as compared to the binary TC because it decodes two bits at the same number of turbo iterations. Double binary TC is less sensitive to puncturing so we can easily use unequal error protection (UEP) for scalable bitstream with CRC bits. More protection is applied to the important part of the bitstream and higher channel code rate is set for data with lower priority. The lowest quality layer which contains the most important data (header, motion vectors and allocation tables) is protected with the lowest channel code rate and vice versa [1]. The available code rates are ($1/3$, $2/5$, $1/2$, $2/3$, $3/4$, $4/5$ and $6/7$).

At the decoder side if a packet is error-corrupted, the CRC check after channel decoding then we point out the corresponding atom in the bitstream. Let an atom (q_x, t_x, s_x) be corrupted after channel decoding or fails to qualify, then CRC checks and remove all the atoms from the bitstream that satisfy the following conditions:

$$q \succ q_x$$
$$t \succ t_x \qquad (2)$$
$$s \succ s_x$$

The new adopted bitstream is then decoded by the SVC decoder. In the channel decoding we use the early stopping (ES) technique. That is, if at some turbo iteration the packet passes the CRC test, there is no need for any further turbo iteration. The ES strongly improves the performance of the channel decoding. The overall performance for the transmitted video is measured as the average PSNR which includes the source coding losses as well as channel error effects. The total PSNR is given as

$$PSNR(R_S) = \max PSNR(R_{SVC}, R_{CRC}, R_{TC}), \qquad (3)$$

where R_S is the overall system rate, R_{SVC} is the rate of the SVC coder, R_{CRC} rate after packetization and R_{TC} is the channel coder rate.

As the packet size is increased, the performance of the double binary TC is improved in terms of bit error rate (BER) and frame error rate (FER). However, if we increase the size of the packet, then there are more atoms in the packet. If a packet fails the CRC check then all the atoms in the packet will be removed from bitstream along with the corresponding atoms satisfying Equation 2. To overcome this problem, we use the almost regular permutation (ARP) [7] interleaver that is better than DVB interleaver with respect to computational cost. To select the best permutation parameters for ARP interleaver, we evaluate the minimum hamming distance (MHD) [4] regarding the each set of permutation parameters. The evaluation is done by M-binary iterative method that produces high amplitude impulse in the decoder for all zero sequence and calculates the minimum distance in an efficient way. For sake of conciseness additional details of M-binary iterative method are not given in this paper.

4. EXPERIMENTAL EVALUATION

Several simulations were performed using the SVC codec, Double binary TC together with QPSK modulation on AWGN channel. Selected results for the FOREMAN CIF sequence at 30fps are reported in this section. Figure 4 shows the average PSNR against the signal to noise ratio Eb/No (Eb represents energy per bit and No is the one sided noise spectral density) for scalable video transmission over an AWGN channel. R_S for equal error protection (EEP) is 500kbps. EEP is applied by using MAP and Max-log-MAP decoding algorithm for channel coding and rate of channel coding R_{TC} is set to $1/2$.

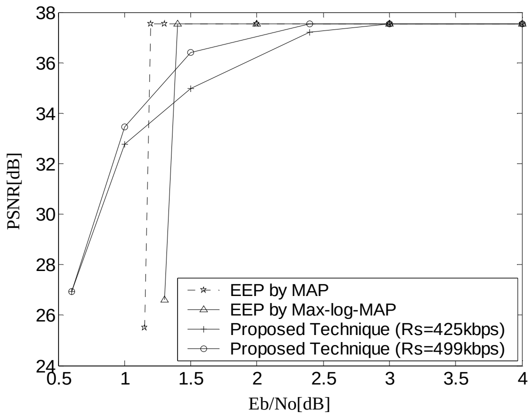

Figure 4: Proposed technique vs Equal Error Protection by MAP and Max-log-MAP decoding algorithm for channel coding.

The results show that there is only 0.15db gain in MAP with respect to Max-log-MAP algorithm for double binary TC but the complexity of MAP is too high with respect to Max-log-MAP. So we use Max-log-MAP for the proposed technique of JSCC because it improves efficiency. In proposed technique, the error protection is applied on a quality layer wise keeping R_S should be less than or equal to 500kbps. The results show that the proposed UEP technique outperforms the EEP, on both R_S (499kbps and 425kbps). Even using such lower overall system rate R_S = 425kbps, the proposed technique shows graceful degradation of the curve at extremely low Eb/No.

The result of the UEP technique with and without ARP interleaver is shown in Figure 5. The best possible ARP interleaver parameters found by M-binary iterative method for length 110 and 188 bytes are $P_0 = 57, P_1 = 7,\ P_2 = 6$, $P_3 = 6$ and $P_0 = 137, P_1 = 5,\ P_2 = 7, P_3 = 5$ respectively.

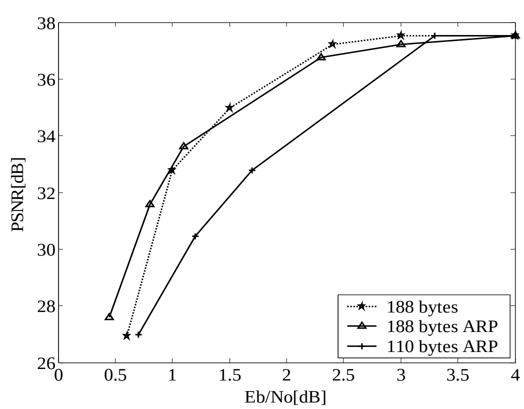

Figure 5: The performance of proposed technique with and without ARP interleaver of length 110 bytes and 188 bytes.

The performance of ARP interleaver is comparable to standard double binary TC interleaver at medium to high signal to noise ratio. But at low signal to noise ratio, the ARP interleaver outperforms the standard interleaver for both lengths 188 bytes and 110 bytes. The performance of ARP interleaver length 110 bytes nearly equal to the 188 bytes without ARP interleaver for UEP at very low Eb/No.

5. CONCLUSIONS

In this paper, we have described an efficient JSCC scheme that combines SVC and double binary TC. The results clearly show that the proposed technique provides a more graceful pattern of quality degradation as compared to EEP at low signal to noise ratio. The ARP interleaver also improves the overall performance. In further work, we will investigate the best trade off between BER and PSNR with respect to the packet size.

6. ACKNOWLEDGEMENT

This research was partially supported by the European Commission under contract FP6-001765 aceMedia.

7. REFERENCES

[1] L. P. Kondi, F. Ishtiaq, A. K. Katsaggelos, “Joint source-channel coding for motion-compensated DCT-Based SNR scalable video”, *IEEE Trans. On image processing*, Vol. 11, No. 9 pp. 1043 – 1052, Sept. 2002.

[2] N. Sprljan, M. Mrak, G. C. K. Abhayaratne, E. Izquierdo. “A scalable coding framework for efficient video adaptation”, *Proc. Int. Work. on Image Analysis for Multimedia Interactive Services WIAMIS*, April 2005.

[3] C. Berrou, A. Glavieux, “Near-optimum error-correcting coding and decoding: Turbocodes.”, *IEEE Transc. Commun.*, Vol. 44, No. 10, pp. 1261-1271, Oct. 1996.

[4] C. Douillard, and C. Berrou. “Turbo codes with rate-m/(m + 1) constituent convolutional codes,” *IEEE Trans. On Comm.* Vol. 53, No. 10, pp. 1630-1638, 10 Oct 2005.

[5] P. Robertson, P. Hoeher, and E. Villeburn. Optimal and suboptimal maximum a Posterioi algorithms suitable for turbo decoding. *Euro Tr, Telecomm.* 8: 119-125, Mar.-Apr. 1997.

[6] T. Zgaljic, N. Sprljan and E. Izquierdo, “Bitstream syntax description based adaptation of scalable video”, *Integration of Knowledge, Semantics and Digital Media Technology (EWIMT 2005)*, pp. 173-176, 30 Nov. 2005.

[7] C. Berrou, Y. Saouter, C. Douillard, S. Kerouédan and M. Jézéquel, “Designing good permutations for turbo codes: towards a single model,” in *Proc. IEEE Int. Conf. Communications*, Paris, France, June 2004.